```
SSSSSSSSSSSS      000000000      RRRRRRRRRRR      TTTTTTTTTTTTTTT     333333333        222222222
SSSSSSSSSSSS      000000000      RRRRRRRRRRR      TTTTTTTTTTTTTTT     333333333        222222222
SSSSSSSSSSSS      000000000      RRRRRRRRRRR      TTTTTTTTTTTTTTT     333333333        222222222
SSS            000      000      RRR      RRR          TTT            333      333      222      222
SSS            000      000      RRR      RRR          TTT            333      333      222      222
SSS            000      000      RRR      RRR          TTT            333      333      222      222
SSS            000      000      RRR      RRR          TTT                     333             222
SSS            000      000      RRR      RRR          TTT                     333             222
SSSSSSSSS      000      000      RRRRRRRRRRR          TTT                   333             222
SSSSSSSSS      000      000      RRRRRRRRRRR          TTT                 333             222
SSSSSSSSS      000      000      RRRRRRRRRRR          TTT               333             222
        SSS    000      000      RRR    RRR           TTT            333             222
        SSS    000      000      RRR    RRR           TTT            333             222
        SSS    000      000      RRR    RRR           TTT          333             222
        SSS    000      000      RRR      RRR         TTT       333      333      222
        SSS    000      000      RRR      RRR         TTT       333      333      222
        SSS    000      000      RRR      RRR         TTT       333      333      222
SSSSSSSSSSSS      000000000      RRR      RRR         TTT       333333333        2222222222222
SSSSSSSSSSSS      000000000      RRR      RRR         TTT       333333333        2222222222222
SSSSSSSSSSSS      000000000      RRR      RRR         TTT       333333333        2222222222222
```

FILEID**SOROUTPUT

```
SSSSSSSS     000000    RRRRRRR      000000   UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
SSSSSSSS     000000    RRRRRRRR     000000   UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
SS          00    00   RR     RR   00    00  UU      UU      TT      PP      PP  UU      UU      TT
SS          00    00   RR     RR   00    00  UU      UU      TT      PP      PP  UU      UU      TT
SS          00    00   RP     RR   00    00  UU      UU      TT      PP      PP  UU      UU      TT
   SSSSSS   00    00   RRRRRRRR    00    00  UU      UU      TT      PPPPPPPP    UU      UU      TT
   SSSSSS   00    00   RRRRRRRR    00    00  UU      UU      TT      PPPPPPPP    UU      UU      TT
       SS   00    00   RR   RR     00    00  UU      UU      TT      PP          UU      UU      TT
       SS   00    00   RR   RR     00    00  UU      UU      TT      PP          UU      UU      TT
       SS   00    00   RR    RR    00    00  UU      UU      TT      PP          UU      UU      TT
       SS   00    00   RR    RR    00    00  UU      UU      TT      PP          UU      UU      TT    ....
SSSSSSSS     000000    RR     RR    000000   UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT    ....
SSSSSSSS     000000    RR     RR    000000   UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT    ....

LL           IIIIII    SSSSSSSS
LL           IIIIII    SSSSSSSS
LL             II    SS
LL             II    SS
LL             II    SS
LL             II      SSSSSS
LL             II      SSSSSS
LL             II            SS
LL             II            SS
LL             II            SS
LL             II            SS
LLLLLLLLLL   IIIIII    SSSSSSSS
LLLLLLLLLL   IIIIII    SSSSSSSS
```

```
0000          1          .TITLE  SOR$OUTPUT                 Output text
0000          2          .IDENT  'V04-000'                  ; File: SOROUTPUT.MAR Edit: PDG016
0000          3          .SBTTL  Copyright Notice
0000          4  ;
0000          5  ;*********************************************************************
0000          6  ;*                                                                   *
0000          7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
0000          8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
0000          9  ;*  ALL RIGHTS RESERVED.                                             *
0000         10  ;*                                                                   *
0000         11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000         12  ;*  ONLY IN  ACCORDANCE WITH  THE   TERMS   OF   SUCH   LICENSE   AND WITH THE  *
0000         13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY   OTHER  *
0000         14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000         15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE   SOFTWARE IS  HEREBY  *
0000         16  ;*  TRANSFERRED.                                                      *
0000         17  ;*                                                                   *
0000         18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000         19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000         20  ;*  CORPORATION.                                                      *
0000         21  ;*                                                                   *
0000         22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000         23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.           *
0000         24  ;*                                                                   *
0000         25  ;*                                                                   *
0000         26  ;*********************************************************************
0000         27  ;
```

```
0000    29                    .SBTTL  Program description
0000    30  ;++
0000    31  ;
0000    32  ; FACILITY:
0000    33  ;
0000    34  ;       Output text by using $PUTMSG.
0000    35  ;
0000    36  ; ABSTRACT:
0000    37  ;
0000    38  ;       NONE
0000    39  ;
0000    40  ; ENVIRONMENT:
0000    41  ;
0000    42  ;       Native mode, User mode, AST reentrant
0000    43  ;
0000    44  ; AUTHOR:
0000    45  ;
0000    46  ;       Peter D Gilbert, June 1982
0000    47  ;
0000    48  ; MODIFIED BY:
0000    49  ;
0000    50  ;       T03-016 PDG016  25-Aug-1982
0000    51  ;               Add a compile-time variable indicating whether to use one
0000    52  ;               $PUTMSG per line, or to use as few $PUTMSG calls as possible.
0000    53  ;       T03-015         Original
0000    54  ;--
```

```
SOR$OUTPUT                 Output text                    16-SEP-1984 01:18:42  VAX/VMS Macro V04-00     Page   3
V04-000                    Program description             5-SEP-1984 03:37:06  [SORT32.SRC]SOROUTPUT.MAR;1      (2)

                          0000    56              .DSABL  GLOBAL               ; Externals must be explicitly declared
                          0000    57              .EXTRN  SYS$FAOL             ; System service to format ASCII text
                          0000    58              .EXTRN  SYS$PUTMSG           ; System service to output a message
                          0000    59              $SHRDEF                      ; Define SHR$_TEXT
                          0000    60
             00000001     0000    61  OPT =       1                           ; If non-zero, minimize $PUTMSG calls
                          0000    62                                          ; If zero, use one $PUTMSG per line
                          0000    63
             00000000     0000    64              .PSECT  SOR$RO_CODE   NOVEC,NOWRT,RD,EXE,SHR,LCL,REL,CON,PIC,LONG
                          0000    65
             00000004     0000    66  CTRSTR =    4                           ; Offset from AP of control string
             00000008     0000    67  PRMLST =    8                           ; Offset from AP of parameter list
             000000FF     0000    68  K_PUTMSG =  255                         ; Maximum size for $PUTMSG
             00000400     0000    69  K_BUFSIZ =  1024                        ; Buffer size
             FFFFFFF8     0000    70  DSC =       -8                          ; Offset from FP of descriptor
             FFFFFFF8     0000    71  LEN =       -8                          ; Offset from FP of length
             FFFFFFFC     0000    72  ADR =       -4                          ; Offset from FP of address
             0A 0D        0000    73  CRLF:       .BYTE   ^X0D, ^X0A          ; Carraige-return, Line-feed
                          0002    74
             003C         0002    75              .ENTRY  SOR$$OUTPUT, ^M<R2,R3,R4,R5>
                          0004    76
                          0004    77  ; Allocate a descriptor and a buffer on the stack
                          0004    78
        FBF8 CE   9F      0004    79              PUSHAB  -K_BUFSIZ-8(SP)     ; Push buffer address
    00000400 8F   DD      0008    80              PUSHL   #K_BUFSIZ           ; Push buffer length
       5E   04 AE   D0    000E    81              MOVL    4(SP), SP           ; Allocate the buffer
          08 AC   9F      0012    82              PUSHAB  PRMLST(AP)          ; Push address of parameter list
          F8 AD   9F      0015    83              PUSHAB  DSC(FP)             ; Push address of buffer descriptor
          F8 AD   9F      0018    84              PUSHAB  LEN(FP)             ; Push address for returned length
          04 AC   DD      001B    85              PUSHL   CTRSTR(AP)          ; Push address of CTRSTR descriptor
    00000000'GF   04 FB   001E    86              CALLS   #4, G^SYS$FAOL      ; Format the output
          7A 50   E9      0025    87              BLBC    R0, 99$            ; Check status
                          0028    88
                          0028    89  ; Form the message vector on the stack
                          0028    90
          F8 AD   9F      0028    91              PUSHAB  DSC(FP)             ; Push address of descriptor
             01   DD      002B    92              PUSHL   #^X00000001         ; Push FAO count, Message flags
       00091130 8F   DD   002D    93              PUSHL   #9@16+SHR$_TEXT     ; Push message ID
       00010003 8F   DD   0033    94              PUSHL   #^X00010003         ; Push count of longwords, Flags (only message)
                          0039    95
                          0039    96  ; Form the argument list on the stack
                          0039    97
             7E   7C      0039    98              CLRQ    -(SP)               ; Two null parameters
             7E   D4      003B    99              CLRL    -(SP)               ; One null parameter
          0C AE   9F      003D   100              PUSHAB  12(SP)              ; Address of message vector
             04   DD      0040   101              PUSHL   #4                  ; Number of arguments
                          0042   102
                          0042   103  ; See whether one $PUTMSG will suffice
                          0042   104
             54   D4      0042   105  10$:        CLRL    R4                  ; Clear the remaining length
          00000001        0044   106              .IF     NE, OPT
    00FF 8F   F8 AD   B1  0044   107              CMPW    LEN(FP), #K_PUTMSG  ; Short enough for one $PUTMSG?
             43   15      004A   108              BLEQ    90$                 ; Branch if short enough
       53   FC AD   D0    004C   109              MOVL    ADR(FP), R3         ; Get address of string
       52   000000FF 8F   D0 0050  110            MOVL    #K_PUTMSG, R2       ; Get maximum length
             54   52   7D 0057   111  20$:        MOVQ    R2, R4              ; Save descriptor of remaining bytes
    63   52   A2 AF   02 39 005A  112              MATCHC  #2, CRLF, R2, (R3)  ; Find a CRLF
```

```
                    F5    13   0060   113          BEQL     20$                        ; Branch if found
          FC AD     55    D1   0062   114  30$:    CMPL     R5, ADR(FP)                ; Same address?
                    11    13   0066   115          BEQL     40$                        ; Branch if no CRLF found
       50   55  FC AD     C3   0068   116          SUBL3    ADR(FP), R5, R0            ; Compute length to output (plus 2)
       54   F8 AD  50     C3   006D   117          SUBL3    R0, LEN(FP), R4            ; Compute remaining length
    F8 AD  50     02      C3   0072   118          SUBL3    #2, R0, LEN(FP)            ; Compute length to output
                    16    11   0077   119          BRB      90$                        ; Go output the text
                          0079   120          .ENDC
                    00000001   0079   121          .IF      EQ, OPT
                          0079   122          MOVL     ADR(FP), R5
                          0079   123          .ENDC
    65   F8 AD   83 AF  02   39   0079   124  40$:    MATCHC   #2, CRLF, LEN(FP), (R5) ; Find first CRLF anywhere
               F8 AD  52   C2   0080   125          SUBL2    R2, LEN(FP)                ; Subtract the remaining length
               F8 AD  02   C2   0084   126          SUBL2    #2, LEN(FP)                ; And two bytes for the CRLF
               F8 AD  50   C0   0088   127          ADDL2    R0, LEN(FP)                ; Add two bytes back if CRLF not found
                    54   52   7D   008C   128          MOVQ     R2, R4                  ; Save string descriptor of remainder
    00000000'GF   6E   FA   008F   129  90$:    CALLG    (SP), G^SYS$PUTMSG            ; Put out the message
               09   50   E9   0096   130          BLBC     R0, 99$                      ; Check status
       F8 AD   54   7D   0099   131          MOVQ     R4, DSC(FP)                       ; Move remainder into the descriptor
          F8 AD   D5   009D   132          TSTL     LEN(FP)                            ; Is remaining length zero?
                    A0   12   00A0   133          BNEQ     10$                          ; No, go output some more
                    04   00A2   134  99$:    RET                                         ; Return
                          00A3   135
                          00A3   136          .END
```

```
ADR             = FFFFFFFC
CRLF              00000000 R      02
CTRSTR          = 00000004
DSC             = FFFFFFF8
K_BUFSIZ        = 00000400
K_PUTMSG        = 000000FF
LEN             = FFFFFFF8
OPT             = 00000001
PRMLST          = 00000008
SHR$_TEXT       = 00001130
SOR$$OUTPUT       00000002 RG     02
SYS$FAOL          ******** X      00
SYS$PUTMSG        ******** X      00
```

+-----------------+
! Psect synopsis !
+-----------------+

PSECT name	Allocation		PSECT No.		Attributes									
. ABS .	00000000	(0.)	00	(0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	00000000	(0.)	01	(1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
SOR$RO_CODE	000000A3	(163.)	02	(2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

+-------------------------+
! Performance indicators !
+-------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.77
Command processing	119	00:00:00.48	00:00:03.59
Pass 1	130	00:00:01.60	00:00:06.09
Symbol table sort	0	00:00:00.14	00:00:00.51
Pass 2	42	00:00:00.40	00:00:01.33
Symbol table output	3	00:00:00.02	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	327	00:00:02.73	00:00:12.35

The working set limit was 1050 pages.
7520 bytes (15 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 121 non-local and 6 local symbols.
136 source lines were read in Pass 1, producing 13 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

+---------------------------+
! Macro library statistics !
+---------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	4

168 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

SOR$OUTPUT   Output text   16-SEP-1984 01:18:42 VAX/VMS Macro V04-00   Page   6
VAX-11 Macro Run Statistics   5-SEP-1984 03:37:06 [SORT32.SRC]SOROUTPUT.MAR;1   (2)

MACRO/DISABLE=TRACE/LIS=LIS$:SOROUTPUT/OBJ=OBJ$:SOROUTPUT MSRC$:SOROUTPUT/UPDATE=(ENH$:SOROUTPUT)

SORRMSIO
LIS

SORMSG
LIS

SORSCRIO
LIS

SOROUTPUT
LIS

SORLIB
LIS